

THESE ARE ERRORS:

```
56.0f,96.0f, 96.1f,96.2f,96.3f, 40.1f, 40.1f,40.2f,40.2f  
states
```

```
07774544Writing Standalone: 40.2  
019238475Writing Standalone: 40.2  
[56.0, 96.0->96.3, 40.1, 40.2, 40.2]
```

1

ERROR IN THIS SECTION

```
56.0f,96.0f, 96.1f,96.2f,96.3f, 40.1f, 40.3f,40.2f,40.2f
```

```
[56.0, 96.2->96.3, 40.1, 40.2, 40.2]
```

2

ERROR IN THIS SECTION

FIXING ERROR 1:

```

CHECKING: 40.1 with 40.2
START: 40.1
56.0f,96.0f, 96.1f,96.2f,96.3f, 40.1f, 40.1f,40.2f,40.2f

40.1
40.199997
40.0
TRACK1
K1=nums.length-2
*****WRITTEN END-----: 40.1
This is counter at the moment: 0
It is not possible to trigger hasTransition if counter is 0 since can not see transition in opposite direction
COUNTER NOT EQUAL TO 0
NOT DESCENDING SEQUENCE-----
TEMP IS BLANK
LAST ITEM SMALLER OR NEXT ITEM BIGGER
NEXT ITEM IS BIGGER
NOT isFirstOccurrenceAscendingChainNoTransition
WHEN-----
07774544Writing Standalone: 40.2
019238475Writing Standalone: 40.2
[56.0, 96.0->96.3, 40.1, 40.2, 40.2]

```

We are in section of code which states previous number smaller next number greater

This is ok

In this gap here, we would have expected it to utilize the store. However there are only instances of this when k==0 or prev number greater(difference) and next number greater (difference). This suggests I need to implement code to use the store again....

It is here due to k==nums.length-2 and nums[k]== nums[k+1]

```

830
831
832 I have introduced this
833 new code and as
834 always, once I perform
835 the standalone write
836 due to duplication, I
837 need to ensure it writes
838 value in the chain
839 40.1->40.2
840 And if there are stored
841 items, then we know we
842 are not to write the
843 standalone twice...
844 We only write it once..
845
846
847
848
849

```

```

if (!isFirstOccurrenceAscendingChainNoTransition)
{
    System.out.println("NOT isFirstOccurrenceAscendingChainNoTransition");
    start=String.valueOf(nums[k]);

    isFirstOccurrenceAscendingChainNoTransition=true;
}

if (isFirstOccurrenceAscendingChainNoTransition)
{
    //ascendency, this is simply used on the basis of consecutive ascending numbers

    System.out.println("CURRENT START: " + start);
    start=String.valueOf(nums[k]);
    end=String.valueOf(nums[k+1]);

    potentialFurtherAscendingBeyondThisStart = start;
    potentialFurtherAscendingBeyondThisEnd = end;
    System.out.println("-----225Stored start -> end: " + start + "->" + end);
}

```

```

1053
1054 {
1055     if (k==(nums.length-2))
1056     {
1057         if (!potentialFurtherAscendingBeyondThisStart.equals("") && !potentialFurtherAscendingBeyondThisEnd.equals(""))
1058             && isFirstOccurrenceAscendingChainNoTransition
1059         {
1060             sm.add(potentialFurtherAscendingBeyondThisStart+"->" + nums[k]);
1061             System.out.println("-----23229USING STORED TO WRITE RANGE");
1062             System.out.println("9705Writing range: " + potentialFurtherAscendingBeyondThisStart + "->" + potentialFurtherAscendingBeyondThisEnd);
1063             potentialFurtherAscendingBeyondThisStart="";
1064             potentialFurtherAscendingBeyondThisEnd="";
1065             System.out.println("CURRENT LIST: " + sm);
1066         }
1067         else
1068         {
1069             start = String.valueOf(nums[k]);
1070             sm.add(start);
1071             System.out.println("07774544Writing Standalone: " + start);
1072         }
1073     }
1074     start = String.valueOf(nums[k]);

```

I narrowed this down as much as possible.

[56.0, 96.2->96.3, 40.1, 40.1->40.2, 40.2]

Infact we can see that it has cut off. This is because it has been overwritten by 96.2->96.3 since the above loop

56.0f,96.0f, 96.1f,96.2f,96.3f, 40.1f,

states

```
if (isFirstOccurenceAscendingChainNoTransition)
{
    //ascendency, this is simply used on the basis of consecutive ascending numbers

    System.out.println("CURRENT START: " + start);
    start=String.valueOf(nums[k]);
    end=String.valueOf(nums[k+1]);

    potentialFurtherAscendingBeyondThisStart = start;
    potentialFurtherAscendingBeyondThisEnd = end;
    System.out.println("-----22Stored start -> end: " + start + "->" + end);
}
```

Since this loop did not check storage variables, it had overwritten it... So I required a change in the loop

```
870
871
872
873
874
875
if (isFirstOccurenceAscendingChainNoTransition
&& potentialFurtherAscendingBeyondThisStart.equals("")
&& potentialFurtherAscendingBeyondThisEnd.equals(""))
{
    //ascendency, this is simply used on the basis of consecutive ascending numbers
}
```

012345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697989910010110210310410510610710810911011111211311411511611711811912012112212312412512612712812913013113213313413513613713813914014114214314414514614714814915015115215315415515615715815916016116216316416516616716816917017117217317417517617717817918018118218318418518618718818919019119219319419519619719819920020120220320420520620720820921021121221321421521621721821922022122222322422522622722822923023123223323423523623723823924024124224324424524624724824925025125225325425525625725825926026126226326426526626726826927027127227327427527627727827928028128228328428528628728828929029129229329429529629729829930030130230330430530630730830931031131231331431531631731831932032132232332432532632732832933033133233333433533633733833934034134234334434534634734834935035135235335435535635735835936036136236336436536636736836937037137237337437537637737837938038138238338438538638738838939039139239339439539639739839940040140240340440540640740840941041141241341441541641741841942042142242342442542642742842943043143243343443543643743843944044144244344444544644744844945045145245345445545645745845946046146246346446546646746846947047147247347447547647747847948048148248348448548648748848949049149249349449549649749849950050150250350450550650750850951051151251351451551651751851952052152252352452552652752852953053153253353453553653753853954054154254354454554654754854955055155255355455555655755855956056156256356456556656756856957057157257357457557657757857958058158258358458558658758858959059159259359459559659759859960060160260360460560660760860961061161261361461561661761861962062162262362462562662762862963063163263363463563663763863964064164264364464564664764864965065165265365465565665765865966066166266366466566666766866967067167267367467567667767867968068168268368468568668768868969069169269369469569669769869970070170270370470570670770870971071171271371471571671771871972072172272372472572672772872973073173273373473573673773873974074174274374474574674774874975075175275375475575675775875976076176276376476576676776876977077177277377477577677777877978078178278378478578678778878979079179279379479579679779879980080180280380480580680780880981081181281381481581681781881982082182282382482582682782882983083183283383483583683783883984084184284384484584684784884985085185285385485585685785885986086186286386486586686786886987087187287387487587687787887988088188288388488588688788888989089189289389489589689789889990090190290390490590690790890991091191291391491591691791891992092192292392492592692792892993093193293393493593693793893994094194294394494594694794894995095195295395495595695795895996096196296396496596696796896997097197297397497597697797897998098198298398498598698798898999099199299399499599699799899910001001100210031004100510061007100810091010101110121013101410151016101710181019102010211022102310241025102610271028102910301031103210331034103510361037103810391040104110421043104410451046104710481049105010511052105310541055105610571058105910601061106210631064106510661067106810691070107110721073107410751076107710781079108010811082108310841085108610871088108910901091109210931094109510961097109810991100110111021103110411051106110711081109111011111112111311141115111611171118111911201121112211231124112511261127112811291130113111321133113411351136113711381139114011411142114311441145114611471148114911501151115211531154115511561157115811591160116111621163116411651166116711681169117011711172117311741175117611771178117911801181118211831184118511861187118811891190119111921193119411951196119711981199120012011202120312041205120612071208120912101211121212131214121512161217121812191220122112221223122412251226122712281229123012311232123312341235123612371238123912401241124212431244124512461247124812491250125112521253125412551256125712581259126012611262126312641265126612671268126912701271127212731274127512761277127812791280128112821283128412851286128712881289129012911292129312941295129612971298129913001301130213031304130513061307130813091310131113121313131413151316131713181319132013211322132313241325132613271328132913301331133213331334133513361337133813391340134113421343134413451346134713481349135013511352135313541355135613571358135913601361136213631364136513661367136813691370137113721373137413751376137713781379138013811382138313841385138613871388138913901391139213931394139513961397139813991400140114021403140414051406140714081409141014111412141314141415141614171418141914201421142214231424142514261427142814291430143114321433143414351436143714381439144014411442144314441445144614471448144914501451145214531454145514561457145814591460146114621463146414651466146714681469147014711472147314741475147614771478147914801481148214831484148514861487148814891490149114921493149414951496149714981499150015011502150315041505150615071508150915101511151215131514151515161517151815191520152115221523152415251526152715281529153015311532153315341535153615371538153915401541154215431544154515461547154815491550155115521553155415551556155715581559156015611562156315641565156615671568156915701571157215731574157515761577157815791580158115821583158415851586158715881589159015911592159315941595159615971598159916001601160216031604160516061607160816091610161116121613161416151616161716181619162016211622162316241625162616271628162916301631163216331634163516361637163816391640164116421643164416451646164716481649165016511652165316541655165616571658165916601661166216631664166516661667166816691670167116721673167416751676167716781679168016811682168316841685168616871688168916901691169216931694169516961697169816991700170117021703170417051706170717081709171017111712171317141715171617171718171917201721172217231724172517261727172817291730173117321733173417351736173717381739174017411742174317441745174617471748174917501751175217531754175517561757175817591760176117621763176417651766176717681769177017711772177317741775177617771778177917801781178217831784178517861787178817891790179117921793179417951796179717981799180018011802180318041805180618071808180918101811181218131814181518161817181818191820182118221823182418251826182718281829183018311832183318341835183618371838183918401841184218431844184518461847184818491850185118521853185418551856185718581859186018611862186318641865186618671868186918701871187218731874187518761877187818791880188118821883188418851886188718881889189018911892189318941895189618971898189919001901190219031904190519061907190819091910191119121913191419151916191719181919192019211922192319241925192619271928192919301931193219331934193519361937193819391940194119421943194419451946194719481949195019511952195319541955195619571958195919601961196219631964196519661967196819691970197119721973197419751976197719781979198019811982198319841985198619871988198919901991199219931994199519961997199819992000200120022003200420052006200720082009201020112012201320142015201620172018201920202021202220232024202520262027202820292030203120322033203420352036203720382039204020412042204320442045204620472048204920502051205220532054205520562057205820592060206120622063206420652066206720682069207020712072207320742075207620772078207920802081208220832084208520862087208820892090209120922093209420952096209720982099210021012102210321042105210621072108210921102111211221132114211521162117211821192120212121222123212421252126212721282129213021312132213321342135213621372138213921402141214221432144214521462147214821492150215121522153215421552156215721582159216021612162216321642165216621672168216921702171217221732174217521762177217821792180218121822183218421852186218721882189219021912192219321942195219621972198219922002201220222032204220522062207220822092210221122122213221422152216221722182219222022212222222322242225222622272228222922302231223222332234223522362237223822392240224122422243224422452246224722482249225022512252225322542255225622572258225922602261226222632264226522662267226822692270227122722273227422752276227722782279228022812282228322842285228622872288228922902291229222932294229522962297229822992300230123022303230423052306230723082309231023112312231323142315231623172318231923202321232223232324232523262327232823292330233123322333233423352336233723382339234023412342234323442345234623472348234923502351235223532354235523562357235823592360236123622363236423652366236723682369237023712372237323742375237623772378237923802381238223832384238523862387238823892390239123922393239423952396239723982399240024012402240324042405240624072408240924102411241224132414241524162417241824192420242124222423242424252426242724282429243024312432243324342435243624372438243924402441244224432444244524462447244824492450245124522453245424552456245724582459246024612462246324642465246624672468246924702471247224732474247524762477247824792480248124822483248424852486248724882489249024912492249324942495249624972498249925002501250225032504250525062507250825092510251125122513251425152516251725182519252025212522252325242525252625272528252925302531253225332534253525362537253825392540254125422543254425452546254725482549255025512552255325542555255625572558255925602561256225632564256525662567256825692570257125722573257425752576257725782579258025812582258325842585258625872588258925902591259225932594259525962597259825992600260126022603260426052606260726082609261026112612261326142615261626172618261926202621262226232624262526262627262826292630263126322633263426352636263726382639264026412642264326442645264626472648264926502651265226532654265526562657265826592660266126622663266426652666266726682669267026712672267326742675267626772678267926802681268226832684268526862687268826892690269126922693269426952696269726982699270027012702270327042705270627072708270927102711271227132714271527162717271827192720272127222723272427252726272727282729273027312732273327342735273627372738273927402741274227432744274527462747274827492750275127522753275427552756275727582759276027612762276327642765276627672768276927702771277227732774277527762777277827792780278127822783278427852786278727882789279027912792279327942795279627972798279928002801280228032804280528062807280828092810281128122813281428152816281728182819282028212822282328242825282628272828282928302831283228332834283528362837283828392840284128422843284428452846284728482849285028512852285328542855285628572858285928602861286228632864286528662867286828692870287128722873287428752876287728782879288028812882288328842885288628872888288928902891289228932894289528962897289828992900290129022903290429052906290729082909291029112912291329142915291629172918291929202921292229232924292529262927292829292930293129322933293429352936293729382939294029412942294329442945294629472948294929502951295229532954295529562957295829592960296129622963296429652966296729682969297029712972297329742975297629772978297929802981298229832984298529862987298829892990299129922993299429952996299729982999300030013002300330043005300630073008300930103011301230133014301530163017301830193020302130223023302430253026302730283029303030313032303330343035303630373038303930403041304230433044304530463047304830493050305130523053305430553056305730583059306030613062306330643065306630673068306930703071307230733074307530763077307830793080308130823083308430853086308730883089309030913092309330943095309630973098309931003101310231033104310531063107310831093110311131123113311431153116311731183119312031213122312331243125312631273128312931303131313231333134313531363137313831393140314131423143314431453146314731483149315031513152315331543155315631573158315931603161316231633164316531663167316831693170317131723173317431753176317731783179318031813182318331843185318631873188318931903191319231933194319531963197319831993200320132023203320432053206320732083209321032113212321332143215321632173218321932203221322232233223432253226322732283229323032313232323332343235323632373238323932403241324232433244324532463247324832493250325132523253325432553256325732583259326032613262326332643265326632673268326932703271327232733274327532763277327832793280328132823283328432853286328732883289329032913292329

FIXING ERROR 2:

```
CHECKING: 40.3 with 40.2
START: 40.1

40.1
40.399998
40.2
HEREEEEE
COUNTER IS 0-----
HERE!!!!
IN HERE!!!!
40.3
40.2

Establishing start: 40.3
WHEN-----
07774544Writing Standalone: 40.2
019238475Writing Standalone: 40.2
[56.0, 96.2->96.3, 40.1, 40.2, 40.2]
```

Similar to test case above, there is no logic implemented to perform a store

56.0f, 96.0f, 96.1f, 96.2f, 96.3f, 40.1f, 40.3f, 40.2f, 40.2f

```

384 //LATE CHANGE IN DOCUMENTATION
385 if (String.valueOf(nums[k+1]).equals(potentialFurtherAscendingBeyondThisStart))
386 {
387     System.out.println("123456using stored start");
388     sm.add(potentialFurtherAscendingBeyondThisStart+"->"+potentialFurtherAscendingBeyondThisEnd);
389     System.out.println("197618writing range: " + potentialFurtherAscendingBeyondThisStart+"->"+potentialFurtherAscendingBeyondThisEnd);
390     //writtenPrevious=false;
391     potentialFurtherAscendingBeyondThisStart="";
392     potentialFurtherAscendingBeyondThisEnd="";
393 }
394
395 //To be 100% sure it does not interfere with above, I will narrow down the loop
396
397 if ((Math.abs(nums[k] - (nums[k+1] - difference)) < epsilon)
398    && potentialFurtherAscendingBeyondThisStart.equals("")
399    && potentialFurtherAscendingBeyondThisEnd.equals(""))
400 {
401     System.out.println("CURRENT START: " + start);
402     start=String.valueOf(nums[k]);
403     end=String.valueOf(nums[k+1]);
404
405     potentialFurtherAscendingBeyondThisStart = start;
406     potentialFurtherAscendingBeyondThisEnd = end;
407     System.out.println("-----Stored start -> end: " + start + "->" + end);
408
409     System.out.println("$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$CURRENT list: " + sm);
410     //writtenPrevious=true;
411     System.out.println("CURRENT START: " + start);
412 }

```

I have added this, and I expect the section of code processing the standalone henceforth to take over from here

[56.0, 96.0->96.3, 40.1, 40.3->40.2, 40.2]

Like always, I will need to revisit all my test cases again....

And it has failed against ChatGPT data in a way that has never appeared before..

So I am taking a small extract to investigate:

20.0f, 19.9f, 19.8f, 63.5f, 14.9f, 14.9f, 48.6f, 48.5f, 48.4f, 48.3f, 48.2f, 48.3f, 48.4f, 48.5f, 48.6f

We can see it has missed out 63.5f, we can see that it has created 20->19.9 which is irrelevant...

```
[20.0->19.8, 20.0->19.9, 14.9, 14.9, 48.6->48.2, 48.2->48.6]
```



```

1051
1052     else //prev number not greater(difference) and next number not greater(difference)
1053     {
1054         //so we know we need to check backwards if less than (within difference)
1055         //we do not need to worry about the
1056
1057         System.out.println("~~~~~6");
1058
1059         if (!(Math.abs(nums[k] - (nums[k+1] + difference)) < epsilon))
1060         {
1061             if (!potentialFurtherAscendingBeyondThisStart.equals(""))
1062                 && !(potentialFurtherAscendingBeyondThisEnd.equals(""))
1063             {
1064                 sm.add(potentialFurtherAscendingBeyondThisStart+"->"+nums[k]);
1065                 System.out.println("-----4444444 STORED TO WRITE RANGE");
1066                 System.out.println("9705Writing range: " + potentialFurtherAscendingBeyondThisStart + "-> " + nums[k]);
1067                 potentialFurtherAscendingBeyondThisStart="";
1068                 potentialFurtherAscendingBeyondThisEnd="";
1069                 System.out.println("CURRENT LIST: " + sm);
1070             }
1071         }
1072     }
1073
1074     else
1075     {
1076         System.out.println("2Writing range: " + start + "-> " + end);
1077
1078         sm.add(start+"->"+end);
1079     }
1080

```

We know we are in descending area of code since counter!=0
So I have checked if the next is descending.. IF not, then it will terminate the range with current nums [k] value.. Otherwise it will process 2writing range as per usual in else statement

[20.0->19.8, 63.5, 14.9, 14.9, 48.6->48.2, 48.2->48.6]

I will now need to go through the ChatGPT data again

Unfortunately as can be seen in the last two rows of the output.xlsx I have landed into all sorts of issues...

I believe these errors were not present before...

So I have rolled back and discarded all the fixes above...

Perhaps I need to put these failed test cases in my code again...

And find a way to implement so it does not violate anything else...

This section of code will be my master test case from now on.. It's a section from ChatGPT data..

There is absolutely no issues with the code last published on my site:

30032025/SummaryRange/8/Final/v1/Solution.java">

85.3f, 85.2f, 19.6f, 19.7f, 19.8f, 19.9f, 20.0f, 19.9f, 19.8f, 63.5f, 14.9f, 14.9f,

[85.3->85.2, 19.6->20.0, 20.0->19.8, 63.5, 14.9, 14.9]

I will now try the above test cases in this code... and start process again..

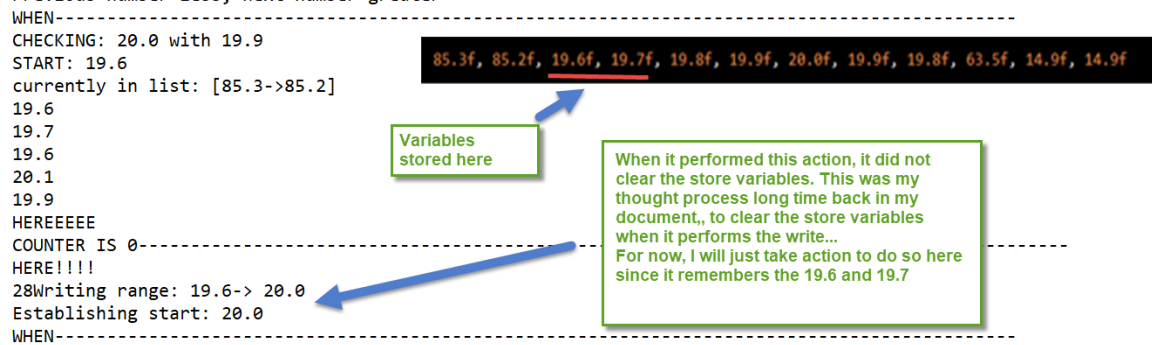
```
85.3f, 85.2f, 19.6f, 19.7f, 19.8f, 19.9f, 20.0f, 19.9f, 19.8f, 63.5f, 14.9f, 14.9f
```

This is the correct state

```
[85.3->85.2, 19.6->20.0, 20.0->19.8, 63.5, 14.9, 14.9]
```

And as soon as I apply the fix for error 1, I finish with:

```
[85.3->85.2, 19.6->20.0, 20.0->19.8, 19.6->19.7, 14.9, 14.9]
```



It if fixed again....

```
[85.3->85.2, 19.6->20.0, 20.0->19.8, 63.5, 14.9, 14.9]
```

I will also make sure the test case associated with error 1 is still functional

TEST CASE 1:

```
56.0f, 96.0f, 96.1f, 96.2f, 96.3f, 40.1f, 40.1f, 40.2f, 40.2f
```

```
[56.0, 96.0->96.3, 40.1, 40.1->40.2, 40.2]
```

All functional

TEST CASE 2:

```
56.0f,96.0f, 96.1f,96.2f,96.3f, 40.1f, 40.3f,40.2f,40.2f //(does not write standalones at end due to skipping iteration)
```

```
[56.0, 96.0->96.3, 40.1, 40.2, 40.2]
```

I have now applied exact same fix as above...

It is beginning to cross my mind since I failed to cleared those store variables it had detriment in lots areas...

And it is fine...

```
[56.0, 96.0->96.3, 40.1, 40.1->40.2, 40.2]
```

So now, I will try the ChatGPT data again.. I hope this is final effort.

It appears that the fix for error 2 is the cause of the issue...

I will need to try and find another workaround for this problem or try to understand why process is failing ...

I have decided to move the location of the code block (used for error 2) into the else statement here

Reason for this choice was that it still facilitates to check for number before and link it all up...

```
1070         if (k==(nums.length-2))
1071         {
1072             //To be 100% sure it does not interfere with above, I will narrow down the loop
1073
1074             System.out.println("*****");
1075             System.out.println(nums[k]);
1076             System.out.println(nums[k-1]);
1077
1078             if ((Math.abs(nums[k] - (nums[k-1] - difference)) < epsilon)
1079                 && potentialFurtherAscendingBeyondThisStart.equals("")
1080                 && potentialFurtherAscendingBeyondThisEnd.equals(""))
1081             {
1082
1083                 System.out.println("CURRENT START: " + start);
1084                 start=String.valueOf(nums[k-1]);
1085                 end=String.valueOf(nums[k]);
1086
1087                 System.out.println("9705Writing range: " + start + "-> " + end);
1088                 sm.add(start+"->"+end);
1089             }
```



```
CHECKING: 40.3 with 40.2
START: 40.1
currently in list: [56.0, 96.0->96.3, 40.1]
```

```
40.1
40.399998
40.2
HEREEEEE
COUNTER IS 0-----
HERE!!!!
IN HERE!!!!
Establishing start: 40.3
```

```
WHEN-----
*****
```

```
40.2
40.3
CURRENT START: 40.3
9705Writing range: 40.3-> 40.2
07774544Writing Standalone: 40.2
019238475Writing Standalone: 40.2
[56.0, 96.0->96.3, 40.1, 40.3->40.2, 40.2, 40.2]
```

It has identified 40.3 is greater than 40.2 and created the chain....

Note if the last number was 40.1, it would not become an issue in this section of code

And now also my ChatGPT code remains ok with no issues.....

But as can be seen above it has written 40.2 twice and also in the merge...

If it is included in the merge, we need to include code to prevent it writing it additional time..

```
830         isPrevNumAscending=true;
831     }
832
833     if (!potentialFurtherAscendingBeyondThisStart.equals("")
834         && !(potentialFurtherAscendingBeyondThisEnd.equals("")))
835     {
836         sm.add(potentialFurtherAscendingBeyondThisStart+"->"+nums[k]);
837         System.out.println("-----23229USING STORED TO WRITE RANGE");
838         System.out.println("9705Writing range: " + potentialFurtherAscendingBeyondThisStart + "-> " + potentialFurtherAscendingBeyondThisEnd);
839         potentialFurtherAscendingBeyondThisStart="";
840         potentialFurtherAscendingBeyondThisEnd="";
841         System.out.println("CURRENT LIST: " + sm);
842     }
843     else
844     {
845         start = String.valueOf(nums[k]);
846         sm.add(start);
847         System.out.println("07774544Writing Standalone: " + start);
848     }
849 }
850
851 if (!isPrevNumAscending)
852 {
853     start = String.valueOf(nums[k]);
854     sm.add(start);
855     System.out.println("019238475Writing Standalone: " + start);
856 }
857 }
858 isPrevNumAscending=false;
859 }
```

It will only perform this if it has not merged num[k] as part of